

Unravelling software when the relationship ends: Doug and Bob's sad story

September 2015

Speed read

Badly written or undocumented software is a big, common problem. The software developer can write the code in a way that may take considerable time and cost for a third party developer to understand and work on it, such as when support is needed, or changes are necessary.

This can leave a company very exposed if the developer of its software is not available (e.g. due to breach of contract, he/she dies or moves overseas, and so on).

With the approach largely "in the developer's head," replacement developers have a big problem translating the Swahili into English. Some companies even have mission critical systems exposed like this, as we've seen in advising clients. So, when they fail...

This problem crops up in a variety of ways. Here, we give a recent court judgment example: Doug McLean and Bob Marshall's terminated JVA where Bob developed the software. When Bob was no longer in the JV, Doug spent large sums to build the software to a satisfactory state.

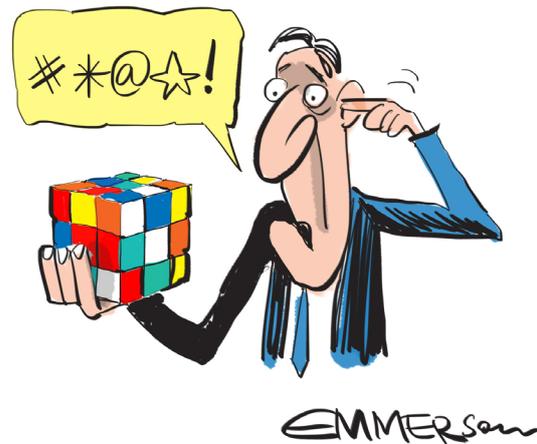
We checked this problem out with Nathan Donaldson, CEO at [Boost Agile](#). He has helpful thoughts on what to do:

"Key to ensuring value from software development effort is adopting current international best practice around Agile & Lean development. These approaches ensure that the software is always in a production ready state (if not actually in production). This includes any documentation that is needed to support the ongoing maintenance and extension of the software.

Documentation is often woefully inadequate, at best describing an unattained and somewhat mystical version of what the software should be. It is imperative to ensure that the documentation describes the current state of the software. The best way to achieve this is to make documentation core to development process. Modern software development techniques like Acceptance Test Driven Development (ATDD) do a great job of documenting and verifying the business logic of the software. Are your software developers and teams using modern approaches to ensure that you are getting the value from their effort?"

So, when it comes to drafting the underlying ICT contracts, it's worth focussing on terms that drive best practice approaches, such as the options that Nathan describes.

SOFTWARE CODE AT TERMINATION DATE



Unravelling software when the relationship ends: Doug and Bob's sad story

The Detail

So, here's the case that has recently been to the Court of Appeal. Doug McLean and Bob Marshall had a JV agreement (JVA), with equal sharing, to develop software to enable automated foreign exchange trading. They were founding members of the Pork Bellies Fun Club, and they felt hindered in managing forex contracts as there was no automated system.

The software system was built mainly by Bob, and was based on an existing US-based internet trading platform. Under the JVA, Bob committed to:

- Progressively provide updated executable versions of the software to Doug; and
- Put the source code for each of those iterations in a location accessible by Doug.

Under the JVA, either party could terminate with three months' notice, which is what Bob did in March 2012. The JVA provided for what happens on termination. Doug and Bob could separately use the software from that point, and the source code had to be available to Doug.

In breach of the agreement, updated source code at and before the date of termination was not provided by Bob, despite numerous requests by Doug. Finally, after about a year from termination, Bob provided the source code to Doug. But it was in a form that was developed well past termination date, to suit Bob's needs.

Doug says he paid a software programmer around \$100K to unravel the software to get it back to where he could use it.

In answer to Doug's claim that Bob should reimburse him the \$100K by way of damages, the High Court said:¹

"The obligation incumbent upon [Bob] Marshall was to supply the source code. Though it was considerably belated, he eventually complied with this request. The difficulties associated with [Doug] McLean and his associates coming to grips with what has been categorised as a very complex piece of software following termination of the Agreement is simply a natural consequence of that divorce. Moreover, the fact that the code was not self-documenting is further evidence of the rapid evolution of the System and that it was never complete. In this respect it has been long recognised that a defendant is not liable in damages for failing to do something of which they are not bound."

In other words, the problem is the difficulty of Doug's programmer getting to grips with complex and undocumented software. There is no duty in the JVA in that regard, so, therefore, no liability. (So far in the story, this tells us to write in specific obligations in contracts to enable the software to be readily usable by others).

The Court of Appeal allowed an appeal on this point as there was a breach of the obligation to provide the code as at termination date, and before. As the Court said:²

"[Bob] Marshall made changes to the source code to suit his own requirements. Those changes would not necessarily have met [Doug] McLean's own needs. Contractually, [Doug] McLean was entitled to call for a copy of the code in its termination date form."

[15] Because the code was not provided in that form, it was necessary for [Doug] McLean to engage a computer consultant to restore it to the form in which it had been on termination."

Unravelling software when the
relationship ends: Doug and
Bob's sad story

So the case now goes back to the High Court to figure out the damages based on breach of the obligation to provide the source code as at the date of termination.

Conclusions

We think that, if this is anything like the problems we have often seen, the software as at termination date would have been very difficult to unravel too, so Doug may not get much of the \$100K paid for by Bob (for the reasons explained in the passage in the High Court Judgment). Often, unless software developers are using best practice a third party, trying to work with the code faces major challenges. The Court of Appeal has decided correctly about the breach of not providing code as at termination date. But it may well be that code at termination date would have led to similar levels of difficulty.

And that puts the non-programmer at a big disadvantage.

Following Nathan's comments above, in drafting the ICT contract we should have in mind the possibility that the current developer is not on the scene and the software needs to be picked up later. Targeted terms are good, not just broader terms such as a requirement to apply best practice.

-
1. McLean v Marshall [2015] NZCA 370 at [14]
 2. McLean v Marshall [2015] NZCA 370 at [14]

Wigley+Company
PO Box 10842
Level 6/23 Waring Taylor Street, Wellington
T +64(4) 472 3023 E info@wigleylaw.com
and in Auckland
T +64(9) 307 5957
www.wigleylaw.com

We welcome your feedback on this article and any enquiries in relation to its contents. This article is intended to provide a summary of the material covered and does not constitute legal advice. We can provide specialist legal advice on the full range of matters contained in this article.